Assemblers can control the parts of your PC that other programming languages can't reach – and these days the development software is friendly too. Dave Jewell picks a winner from the top five packages.

ASSEMBLE FOR ACTION

Assembly language programming has been with us for almost as long as programming itself. The earliest computers were programmed by entering machine code (the low-level sequences of binary ones and zeros, which is all that computers really understand) directly into memory, using a series of switches. However this was incredibly slow, boring and error-prone.

It wasn't long before some bright spark realised that slow, boring tasks are just what computers are good at – so why not get the computer itself to work out the sequences of ones and zeros which make up each instruction?

Next, this ingenious individual hit upon the idea of giving each machine code instruction a symbolic name – programmers then only had to write their code as a sequence of symbolic names and the computer handled the job of translating these symbols into something it could understand. For instance, instead of '1000100111011000', you could simply enter 'mov ax,bx'. Which would you rather type?

CHOOSING A LANGUAGE

Naturally, the computer needs a program to do this translation, and such a translator is called an 'assembler' because it assembles sequences of machine code instructions from the programmer-supplied input. However, confusion can arise as a program intended to be used as input to an assembler is often itself called an 'assembler program'.

Although it involves a certain amount of fuss, there are advantages to using Assembly language: the finished program is often smaller and faster than the equivalent high-level language program and there are things that you can do in Assembler that you can't easily manage with traditional high-level languages, including writing Terminate and Stay Resident (TSR) programs and other specialised system software.

In fact, some will argue that Assembler is the only 'proper' programming language, and that anything else is just a poor substitute. The truth, however, is that although it is just too time-consuming to use universally, there are times when only Assembler will do.

Here we review five PC assemblers, including the ever-popular Microsoft *Macro Assembler*, Borland's *Turbo Assembler*, *Optasm* from SLR Systems, Microsoft *Quick Assembler* and finally *A86*, a well-known shareware assembler.

INSTALLATION AND DOCUMENTATION	
TURBO ASSEMBLER	■■■■■
MICROSOFT MASM	■■■■□
MICROSOFT QUICK ASSEMBLER	■■■□□
SLR OPTASM	■■□□□
A86	■□□□□

The only shareware contender in this review, *A86 Version 3.22*, comes on a single 360K disk. The package is supplied as a large set of disk files, but there is no installation program or documentation provided.

A label on the disk invites you to get started by looking at the contents of the INFO file. However, this file opens with

```
SCAN_ARGS:                              TCOLS:
  0544 MOV SI,080                       + 04EA CALL SCAN_ARGS
  0547 LODSB                              04ED CALL READ_SOURCE
  0548 CBW                                04F0 JE 04FF
  0549 XCHG AX,BX                         04F2 CALL GATHER_PAGE
  054A MOV BIBX+SI1,0FF                   04F5 PUSHF
  054D CALL CHECK_DIGIT                   04F6 CALL OUTPUT_PAGE
  0550 JE E1                              04F9 CALL OFLUSH
  0552 CALL SCAN_DECIMAL                  04FC POPF
  0555 MOV NCOLS,AL                       04FD JNE 04F2
  0558 XCHG AX,BX                         04FF JMP GOOD_EXIT
  0559 MOV AL,6                         CHECK_DIGIT:
  055B CALL SCAN_DECIMAL                  0502 PUSH AX
  055E MOV SKIPCT,AL                      0503 LODSB

AX 0000   i z e   1:
BX 0000  IP 0549  2:
CX 00FF  CS 2D50  3:
DX 2D50  SS 2D50  4:
SI 0081  DS 2D50  5:
DI FFFE  ES 2D50  6:
BP 0882  SP FFFC  1: 04ED
```

● As you can see, the debugger that comes with the *A86* assembler package is a fairly basic affair in comparison with the all-singing, all-dancing offerings of some commercial packages

PROGRAMMING

the words 'If you have enjoyed using this program....' and contains little else. You have to copy the contents of the disk manually to your hard disk, from where you can run a batch file to uncompress the software.

Fortunately, the on-disk documentation is very thorough, amounting to over half a Megabyte of information once it is expanded. It begins with a tutorial example of how to use the assembler, and includes a comprehensive reference section on the program and the 8086 instruction set.

Optasm, like *A86*, doesn't have an installation program. However, the package does have a large user manual, describing its facilities in considerable detail. What it does leave out is the processor instruction set.

This shortcoming is balanced by OptHelp, the aptly-named on-line help system which allows instant access to Assembler commands, detailed instruction and set information. Unfortunately, OptHelp swallows 115K of RAM, which is unacceptable.

MASM (Microsoft's *Macro Assembler)* has an automatic set-up program, which is simply set in motion from the first disk. The set-up program also installs a copy of *CodeView* (Microsoft's screen-orientated debugger) and helpfully asks if you want to run an interactive *CodeView* tutorial.

The Microsoft documentation is also of a high standard and is intended for both the beginner and experienced programmer. There's also a useful pocket-sized book which includes 80286 and 80386 instructions and information on the numeric co-processor chip.

Interestingly, the version of Microsoft *Quick Assembler* reviewed here has *Quick C* bundled with it. It's basically a stripped-down version of *MASM*, without support for the special 80386 processor instructions.

This version is not a separate product, since it's built into the *Quick C* development environment and command-line compiler. However, you aren't forced to use C – you can write your entire program in Assembler.

This development environment has a particularly impressive help system, providing explanations, summaries and detailed examples of every instruction.

An extremely polished installation program with context-sensitive help is included in Borland's *Turbo Assembler*. Like *MASM*, the documentation includes a handy-sized reference guide and a well-indexed *User's Guide*, which assumes little Assembler experience.

The latter contains many example programs, a number of which are included on the disk. *Turbo Assembler* includes a pop-up, memory-resident help system, which takes up less than a quarter of the memory used by OptHelp.

INDUSTRY COMPATIBILITY

MICROSOFT MASM	■■■■■
MICROSOFT QUICK ASSEMBLER	■■■■
TURBO ASSEMBLER	■■■
SLR OPTASM	■■■
A86	■

Microsoft's *MASM* is the acknowledged standard by which others are judged. Although bulky and slower than many of its younger rivals, it continues to be the professional's choice.

Consequently, a large selection of Public Domain PC software comes with full *MASM*-compatible Assembler source code, so it's important that an assembler should have a high degree of compatibility with *MASM*.

Macro handling is an area which often 'sorts out the men from the boys' as far as assemblers are concerned, and a macro facility is offered by all of the assemblers covered here.

It's possible to take a frequently-used sequence of machine code instructions and wrap it into a 'macro definition'. Whenever you specify the name of the macro inside your assembler source file, that same sequence of instructions will be inserted into your code. Unfortunately, things get more complicated than that, and it's possible to have macros that take parameters, call other macros and so forth.

Microsoft has a large file of macro definitions called CMACROS.INC. This represents something of a torture-test of *MASM* compatibility, so we decided to write a test file using CMACROS.INC and see if each of our assemblers could handle it.

The source to the test file is given in *Macro Handling* below – it simply defines a small routine (callable from C or Pascal) which adds two numbers together and returns the result. Don't be deceived by the small size of this source file: it includes nearly 1,200 lines of very complicated macro definitions.

Not surprisingly, *MASM* scored full marks for compatibility. Microsoft *Quick Assembler* also compiled the test routine without a murmur, although it doesn't rate 100 percent *MASM* compatibility because it doesn't assemble 80386 specific instructions.

Turbo Assembler didn't fare quite so

MACRO HANDLING

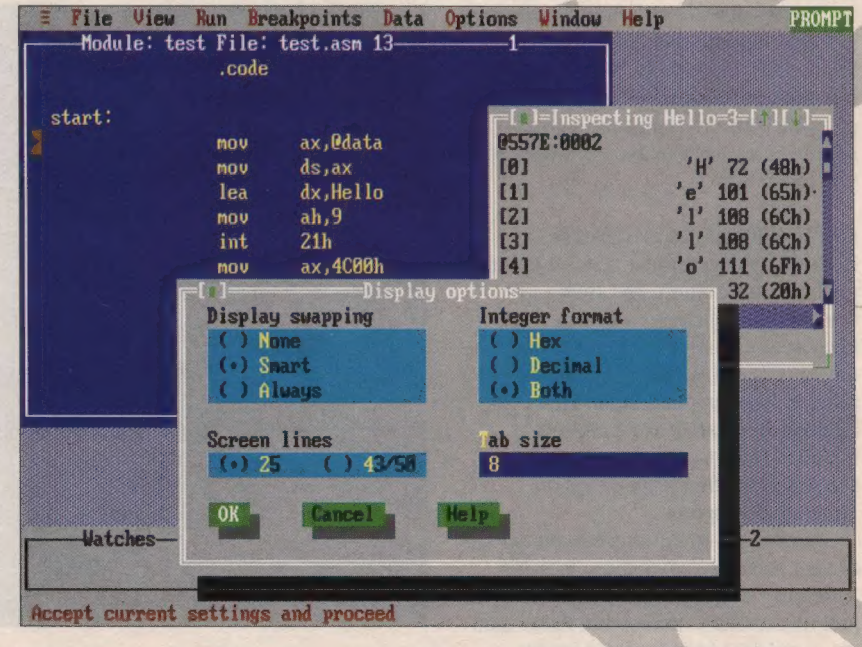
This code sample tests *MASM* macro compatibility – important if an assembler is to be used to compile existing source code.

```
    include cmacros.inc

    sBegin code
    assumes cs,code

cProc AddNumbers,<NEAR,PUBLIC>
    ParmW X
    ParmW Y
cBegin
    mov ax,X
    add ax,Y
cEnd

sEnd code
end
```



● The *Turbo Assembler* package from Borland comes with a pop-up help system that you can call on while writing code in an editor – as seen here with the programmers' editor, *Brief*

HIGH-LEVEL LANGUAGES

Since many applications are written in languages other than Assembler, such as C, Pascal, and C++, it's important that a professional-quality assembler should be able to interface cleanly and easily with a program written in a high-level language. In practice, this means that it should be able to deal with 'high-level' concepts such as structures, stack-based parameters and so on.

Most high-level languages use the processor stack to pass arguments around from one routine to another and a good assembler needs to determine automatically the location of these parameters based on the memory model being used.

An example should serve to make this clearer – suppose we have a data structure composed of two 16-bit integers and a ten byte string, the total size of this data structure is 14 bytes. We may want to write an assembler function (callable from C) which fills the data structure with zeros and returns its size.

To see how a simple function should be implemented, lets look at this file, which tests straightforward *MASM* source code compatibility. It will even assemble without changes under *QuickAssembler*, Microsoft *MASM* (a good thing too) and *Turbo Assembler*.

```
    .model medium,c ; set up model, language

Account struc ; structure definition
Balance dw ? ; current credit
OverDraft dw ? ; allowed overdraft
AccName db 10 dup (?) ; name of account
Account ends ; end of definition

    .code ; start of code

InitAcc proc uses si,Acc:ptr,Bal:word,Over:word

    mov si,Acc ; get ptr to Account
    mov ax,Bal ; get balance
    mov [si].Balance,ax ; put it in account
    mov ax,Over ; get overdraft limit
    mov [si].OverDraft,ax ; put it in account
    add si,AccName ; point to name
    mov cx,10 ; length of name
@@1: mov byte ptr [si],0 ; clear current byte
    inc si ; bump to next
    loop @@1
    mov ax,Type Account ; size of structure
    ret ; back to caller
InitAcc endp

    end
```

```
$CurStructOpen = TRUE                   ;; and current structure
$CurDataItem   = 0                      ;; start with no data items

;; ************************************************
;; RULES FOR METHOD INVOCATIONS ********************
;; ************************************************
;; A method invocation takes the form:
;;   ObjectName_MethodName ObjectParam,AXParm,BXParm,CXParm,DXParm
;;
;; To call the object's ancestor method:
;;   ObjectName_MethodName_A ObjectParam,AXParm,BXParm,CXParm,DXParm
;;
;; Parameters:
;;     ds = segment of ObjectParam unless the method is non
;;        virtual and any parameters are absolutely
;;        unnecessary
;;
;;     some segment register must be in DGROUP with the proper
;;     ASSUME active
;;
;;   ObjectParam = offset of object
;; Action:
;;   defined by method
-- More --
```

● The debugger featured in the Microsoft *Macro Assembler (MASM)* package is the now venerable *CodeView* program. This is still something of a standard despite the rather old fashioned feel to its operation

well. It produced a long list of error messages, even though it has an option which allows it to operate in *MASM*-compatible mode, or use 'Ideal' mode (Borland's preferred approach to Assembler syntax).

There's even a 'Quirks' mode designed to allow *Turbo Assembler* to turn a blind eye to minor inconsistencies in the way *MASM* works. Yet despite all this, we couldn't get Borland's product to assemble the test file successfully.

However, it should be noted that *Turbo Assembler* is generally very compatible with *MASM* – it must be assumed that CMACROS.INC has found a chink in the armour.

Like *Turbo Assembler, Optasm* has a *MASM*-compatibility switch, which allowed the test file to be assembled at a blindingly fast rate. It generated two warning messages, but the object file it produced contained no errors. However, it has other *MASM* incompatibilities, as we shall see in the next section.

A86 was very much an 'also-ran' in the compatibility stakes. Although it doesn't allow you to specify an include file within assembler source, it does allow you to compile a number of files sequentially to achieve the same result.

When we tried compiling the test file, *A86* generated a large number of errors – hardly surprising as it makes no serious attempt to be *MASM* compatible. Although the author of *A86* states in the documentation that he made compatibility a high priority, he unfortunately went on to design a macro language with very different syntax.

He gives much advice on converting *MASM* assembler code to *A86* source, but refers to conversion in the other direction as 'a bit of a blasphemy'.

HIGH-LEVEL LANGUAGE SUPPORT

MICROSOFT MASM	■■■■■
TURBO ASSEMBLER	■■■■■
MICROSOFT QUICK ASSEMBLER	■■■■■
SLR OPTASM	■■■□□
A86	■■□□□

Nowadays, few major programs are written entirely in assembler; while it has the advantages of small size and maximum speed of execution, it takes too long to write an entire application.

However, there are numerous small utilities written this way, and much of the 'systems-level' code on your PC is written entirely in Assembler. Increasingly though, languages such as C, Pascal and C++ are the primary means of development.

This means that a professional-quality assembler should be able to interface simply to a program written in

a high-level language (See *High-Level Languages* page 223). The complexity of this process partly depends on the source code used, and the assemblers reviewed here all use different codes.

The Assembler source code specified in *High Level Language* can be used with *Quick Assembler, Turbo Assembler* and *MASM*. In order to make `InitAcc` into a Pascal callable function (as opposed to C callable) or change the memory model, it's only necessary to alter the `.model` directive. The special `proc` directive names the function and tells the assembler what arguments to expect on the stack.

By contrast, the equivalent source code for use with *Optasm* is subtly different. For example, you can't specify the language you're linking with, which means you have to take care of case-sensitivity, adding a leading underscore to function names and so on.

More significantly though you have to make the function visible to the linker and manually calculate the stack offsets of the passed parameters according to the memory model in use.

Finally, let's consider *A86*. There's little difference between this and the changes for *Optasm* except that *A86* doesn't recognise the `.model` directive and won't accept `add si, AccName`.

To summarise: *Turbo Assembler, Quick Assembler* and *MASM* all have good support for linking to high-level programs. *Optasm* and *A86* are less strong in this area and this, we feel, reflects a difference in philosophy between the two groupings.

The first three include extensive documentation on linking to high-level languages; the other two contain none. If you're planning to use your assembler to develop small, highly-optimised routines for linking into C and Pascal programs, you'd be well advised to go with *Turbo Assembler* or one of the Microsoft products.

However, on the other hand, if your only interest is in writing pure Assembly language programs, then *Optasm* or *A86* could be right for you.

SPECIAL FEATURES

TURBO ASSEMBLER	■■■■■
MICROSOFT QUICK ASSEMBLER	■■■■■
SLR OPTASM	■■■□□
MICROSOFT MASM	■■■□□
A86	■■□□□

An interesting feature of the *Optasm* package is the inclusion of a set of macros (rather like the CMACROS.INC package – mentioned earlier – that comes with the Microsoft *MASM* package) which allow you to write Assembler programs in an object-orientated manner. If, as a devoted assembler programmer, you've been wondering what Object-Oriented Programming (OOP) is all about, here's your chance to find out.

The macros were written by Gibson Research, developers of the popular *Spinrite* disk utility. *Optasm* also includes a simple debugger and a linker, though neither of these are mentioned in the printed documentation.

Optasm has a number of innovative features built into the assembler itself, including the ability to optimise out-of-range jump instructions and push multiple items on to the stack with one instruction. *A86* also has a number of enhancements in this area, including extensions to the AAM and AAD instructions, that allow them to work with a radix other than ten (which exploits a feature of the 80X86 processor that has never been officially documented by Intel).

A86 also includes a debugger and a utility to convert the MAP files generated by the Microsoft *Linker* for use with *A86's* debugger. In addition, there's a source file librarian to group source files into convenient libraries.

A86 doesn't include a linker, but then it doesn't really need one – it generates COM files directly and will automatically assemble other source files held in source libraries when it needs code held in the library.

Turbo Assembler comes bundled with *Turbo Debugger* or as a part of the *Turbo Pascal* and *Turbo C/ Professional Development* systems. As well as being fun to use, the *Turbo Debugger* is generally considered superior to Microsoft's *CodeView* debugger.

Turbo Assembler has an extension to the `.model` directive, which causes it to generate object code compatible with Microsoft *Windows*. The *Turbo Linker* is provided, as well as a MAKE utility, and an object file librarian for maintaining Microsoft-compatible code libraries.

Microsoft's *Macro Assembler* did much to pioneer Assembly language development on the PC, but nowadays is beginning to look a little dated. There's nothing wrong with the assembler, but it doesn't have the speed or polish of *Turbo Assembler*.

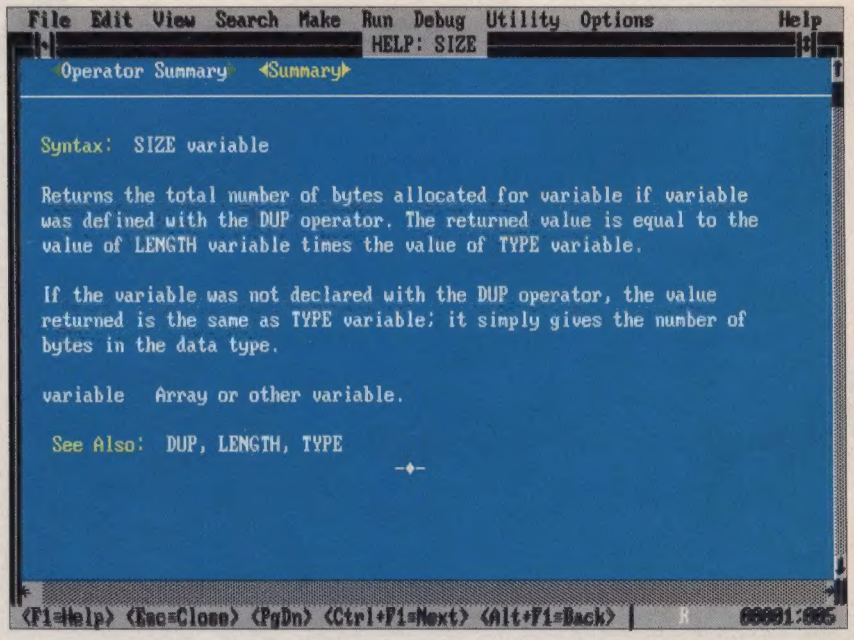
The same is true of the *CodeView*, debugger which is less flexible than the Borland offering and doesn't have such a friendly user interface. *Quick Assembler* is much more interesting, because of it's excellent on-line help system and the integrated environment which it shares with the *Quick C* compiler.

MASM does have one card up its sleeve, however, in the form of OS/2 compatibility. You can use it, together with the other supplied utilities, to create programs that will run under OS/2. Of the various assemblers reviewed here, OS/2 compatibility is unique to *MASM*.

PERFORMANCE AND USAGE

MICROSOFT QUICK ASSEMBLER	■■■■□
SLR OPTASM	■■■■□
TURBO ASSEMBLER	■■■□□
MICROSOFT MASM	■■□□□
A86	■□□□□

Optasm is an exceedingly fast assembler. After installing the program, we decided to try assembling a couple of fairly large files. It appeared that



● The context-sensitive on-line help provided by the Microsoft *Quick Assembler* package is an excellent source of information when writing Assembler in its *Quick C* environment

GET ORGANISED

Personal Accountant for any IBM compatible PC – the most comprehensive personal financial management system ever written – from VITAL Software.

Accounts

- Up to 200 financial accounts and 200 income expense categories
- Accounting, budgeting and project management
- Personal and business accounts fully integrated; any number of users
- Sophisticated standing order processing
- Single entry cashbook accounting, or partial or even full double entry
- Miniature purchase and sales ledger
- All you need for VAT record keeping

Assets & Liabilities

- Up to 200 unit based accounts and up to 200 balance based accounts
- Stores details of all your savings and investments
- Gives complete account history of each of your holdings
- Gives access to a complete personal balance sheet which can include galleries full of pictures, albums full of stamps, cellars full of wine, or indeed, any other asset.
- See your capital gains position on any holding
- Fully integrated with the other modules

Accounts, Inventory & Insurance and Assets & Liabilities can be purchased at £79.95 per module or £149.95 for all three. Prices include postage and VAT.

"I am greatly enjoying Personal Accountant and for once all my accounts are being run on a more secure basis. At least I now have a good idea of where we are! Thank you for your help in these matters. I am very happy to tell all clergy that this is just the kind of thing they need."

Reverend M Worgan

Inventory & Insurance

- Up to 50 buildings & insurance policies, 200 rooms & categories, and up to 250 articles per inventory
- Stores current value, insured value and cost
- Up to six lines of description for each article
- Produces reports in summary or detailed form – ideal for valuers
- Valuations can be inflated each year, either for all articles or for all articles in selected categories
- Produces inventories sorted by buildings, rooms, categories, or insurance policies
- Gives access to a summary of the adequacy of your insurance policies and a twelve month diary showing upcoming premium payments due

New Tax Reckoner (Available July 1991)

Tax Reckoner is an Income and Capital Gains tax package for UK tax-payers. Its specification is virtually that of a professional tax package with the ease of use, and at the price of, a personal package. Store all pieces of tax related information – dividend receipts, Assessments, deductible expenses etc. – as they are received during the tax year, and complete your Tax Return in minutes at the end. Key in the new rates and bands as they are announced at Budget time, or model the effect of changes in your income or expenses, or the composition of your assets, and show the effect of these changes on your tax position. The Tax Reckoner produces all figures for incorporation into the Tax Return together with all supporting schedules. It also produces a tax estimate and a simulated IR form 930. It contains retail price index information and can estimate Capital Gains Tax liability. The rules for 1990/91 and 1991/92 are supplied; rules for subsequent years up to 1999/2000 can be input by the user, though annual updates will be available.
This module costs £119.95 inclusive of postage and VAT.

"Trial without Tears" means return it if you don't like it – money back without question!

. . . has succeeded . . . unique . . .
What Personal Computer

. . . the most ambitious & comprehensive . . .
Independent on Sunday

SYSTEMSTAR
LIMITED

1-3 PARLIAMENT SQUARE HERTFORD SG14 1EX
TELEPHONE: (0992) 500919 FACSIMILE: (0992) 554261

Please send me full details of Personal Accountant:

Name: _____

Address: _____

_____ Tel: _____

something had gone wrong when *Optasm* kept coming straight back to the command line, but it only took a few moments to realise that it really was assembling files at quite an amazingly fast speed.

This turn of speed is *Optasm's* greatest strength, and anyone who has a great deal of Assembler work to do will find *Optasm* paying for itself very quickly in terms of time saved.

A86 is another pleasingly fast assembler package, but not perhaps in *Optasm's* class. A noteworthy feature of *A86* is that it inserts error messages directly into the source file at the appropriate place. You then edit the source file to remove the cause of the errors and *A86* will take care of removing the error messages next time you run it.

Usefully, *A86* does make a back-up copy of your original source file, but having an assembler play around with source code seems rather risky. Nevertheless this is certainly quite a novel approach.

The *Turbo Assembler* and *MASM* products are both command line orientated, able to take a series of switches and a list of one or more source files that are to be assembled. In use, they are both very full-featured, but benefit from the provision of a memory resident utility which includes Assembler instructions.

This is an element which *Quick Assembler* doesn't have because of the on-line help provided by the *Quick C*/Assembler environment. Similarly, with *Quick Assembler*, you don't need to remember a long list of obscure command line switches – you merely set up the required options in a dialogue box and off you go.

Optasm and *Quick Assembler* both rate a high score in this category: the former for speed and the latter for simplicity of use. *MASM* and *Turbo Assembler* are both professional-quality Assembler products, but *Turbo Assembler* has the edge on both features and speed.

CONCLUSION

TURBO ASSEMBLER	■■■■■
MICROSOFT MASM	■■■■
MICROSOFT QUICK ASSEMBLER	■■■
SLR OPTASM	■■■
A86	■

While you may feel that we've been rather unfair in our assessment of the *A86* shareware assembler, in actual fact, all things considered, it's quite an impressive little program. However, from the viewpoint of a professional PC programmer, a reasonable level of *MASM* compatibility and a capacity to easily interface with high-level languages is essential.

A86 is an excellent assembler for the hobbyist or the Assembler-only programmer. It can't, however, be whole-heartedly recommended as a professional-quality tool.

A very good choice for anyone who has a great deal of Assembler programming work to carry out is *Optasm*. It's a professional-quality development package and if your job involves writing large chunks of Assembler code (such as an operating system extensions or device drivers), then *Optasm* would be a very wise choice to make.

The Microsoft *Quick Assembler* package is ideal for a C programmer who wishes to try his hand at the Assembly programming language; the on-line help is really invaluable for novice programmers.

However, *Turbo Assembler* and Microsoft's *MASM* are likely to be the best bet for most programmers; they're not as fast as *Optasm*, but they provide everything you need. If you want to do any OS/2 work, buy *MASM* , but if you want an excellent debugger and the ability to generate *Windows*-compatible code, you would be well-advised to buy *Turbo Assembler*. ●

ASSEMBLERS

	Turbo Assembler	MASM	Optasm	Quick Assembler	A86
SUPPLIER	Borland	Microsoft	Grey Matter	Microsoft	Shareware Marketing
CONTACT	(0734) 321150	(0734) 500741	(0364) 53499	(0734) 500741	(0297) 24089
PRICE (including VAT)	£149	£132	£120	£144	$50 Registration
OS/2 SUPPORT	N	Y	N	N	N
PROTECTED MODE INSTRUCTION	Y	Y	Y	N	N
WINDOWS SUPPORT	Y	N	N	N	N
MASM COMPATIBILITY	Excellent	N/A	Fair	Excellent	Poor
SPEED	Good	Fair	Excellent	Good	Good
LINKER	Y	Y	Y	Y	N (produces COM files directly)
HELP SYSTEM	Y	N	Y	Y	N
DEBUGGER	Y	Y	Y	Y	Y
DISPLAY	T	T	T	T	T
ISSUE DISKS	3.5 or 5.25	3.5 or 5.25	3.5 or 5.25	3.5 or 5.25	T
MINIMUM HARDWARE	HD 512	HD 384	HD 512	HD 640	HD 512
OPTIONAL HARDWARE	M	-	-	M	M